Power-Efficient Focal-Plane Image Representation for Extraction of Enriched Viola-Jones Features

Jorge Fernández-Berni, Laurentiu Acasandrei, Ricardo Carmona-Galán,
Ángel Barriga-Barros, Ángel Rodríquez-Vázquez
Institute of Microelectronics of Seville (IMSE-CNM)
Consejo Superior de Investigaciones Científicas y Universidad de Sevilla
C/ Américo Vespucio s/n, 41092, Seville, Spain
Contact email: berni@imse-cnm.csic.es

***Abstract*—This paper describes the use of a reconfigurable focal-plane processing array in order to achieve an image representation which dramatically reduces the computational load of the Viola-Jones object detection framework. Additionally, such representation provides richer information than the simple sum of pixels within rectangular regions originally defined in this framework. As a result, more elaborated features could be devised to speed up the execution of the subsequent attentional cascade, boosting thus the performance of the whole algorithm. The proposed circuitry has been successfully implemented in a CMOS prototype smart imager. Experimental results are given, demonstrating the suitability of the approach presented to efficiently deliver enriched Viola-Jones features.**

I. Introduction

The Viola-Jones framework [1] constitutes one of the best approaches reported to provide real-time visual object recognition. It is based on the extraction of very simple features across the image which are subsequently analyzed by a cascade of classifiers. These classifiers are previously trained according to the object to be detected, adapting progressively their internal thresholds when successive training images are passed through. Despite its simplicity, this framework still requires a considerable amount of computational and memory resources. During the last few years, numerous efforts have been focused on exploiting the increasing memory and logic capabilities available in FPGAs [2], [3] as well as the highly parallel computation structure of GPUs [4], [5]. When it comes to low-power embedded systems, additional constraints must be introduced on the image resolution [6], [7] or the type of processor operations [8] in order to obtain at least moderate frame rates.

A strategy that, to the best of our knowledge, has not been employed yet to accelerate the feature extraction of the Viola-Jones framework is focal-plane sensing-processing [9]. It is supported by the possibility of integrating photosensors with processing hardware in CMOS technologies. Thus, a processing element (PE) can be located close to each photosensor representing a pixel. Every PE is in turn connected to its immediate neighbors, rendering a processing array suitable for early vision tasks through the Single Instruction Multiple Data (SIMD) paradigm [10]. All the locally interconnected PEs execute the same instruction in parallel but applied to different data. Consequently, a pre-processed image flow is delivered, alleviating the computational load of subsequent digital stages. The performance of the system is improved firstly due to the energy efficiency of the focal-plane processing. Also the clock frequency and memory accesses are significantly reduced as the digital processor does not have to realize now a great deal of repetitive operations over a serialized data flow.

A prototype focal-plane sensor-processor chip based on analog PEs has been recently reported in [11]. The fact of using analog circuitry instead of its digital counterpart implies that, if an ultra low-power and area-efficient implementation is intended, the equivalent resolution of the computations can not go beyond 6-7 bits . Fortunately, such moderate resolution usually suffices for most of low-level image processing tasks. The prototype incorporates the possibility of carrying out any of its processing primitives in independent user-definable rectangular-shape blocks. This is characteristic of the Viola-Jones framework as the features it uses for detection are obtained from rectangular pixel grouping. This grouping is constantly changing while the image is being processed in order to extract as many features as necessary for an accurate object recognition. We will see how one of the primitives of the prototype, block-wise energy computation, fits very well into the Viola-Jones processing scheme while taking advantage of the massively parallel and low-power operation of the chip. Besides, for each group of pixels established, the value of the energy accounts for more information than simply the value of their sum originally formulated in the framework. This enables the definition of new features that make the most of this additional information in order to shorten the decision time of the cascade of classifiers.

II. Feature Extraction

The object detection procedure of the Viola-Jones framework is based on Haar-like features [12] like those of Fig. 1. For each feature, the sum of the pixels within the white rectangles is substracted from the sum of the pixels within the black rectangles. For example, the value of features type (a), F_a, is calculated as:

$$F_a = \sum_i \sum_j W_{ij} - \sum_m \sum_n B_{mn} \qquad (1)$$

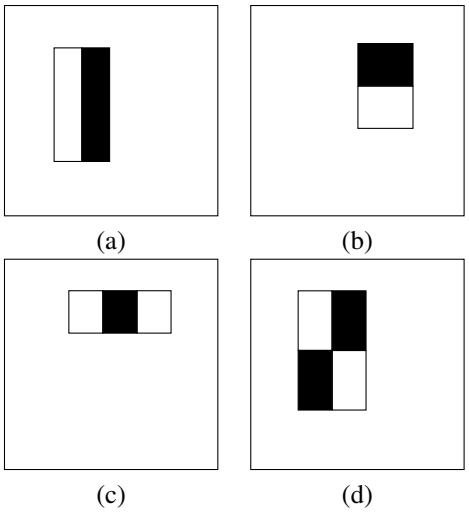

Fig. 1. Rectangle-shape features used in the Viola-Jones framework. They are represented with respect to their enclosing detection window.

where W_{ij} represents the pixel values within the white rectangle and B_{mn} the pixel values for the black rectangle. Note that, in practice, we are simply comparing the DC component of the rectangles involved as the sum of the pixels is proportional to their mean value.

In order to speed up the extraction of features, an intermediate representation called *integral image*, I'_{xy}, must be previously obtained. Each pixel of this representation contains the sum of all the pixels above and to the left of the same pixel in the original image I_{xy}. In this way, any rectangular sum at any scale and location in the latter can be evaluated from only four pixels adequately chosen in the former. Furthermore, the computation of the *integral image* can be realized in one pass over the original image by applying the following recurrences:

$$\begin{cases} S_{xy} = S_{x,y-1} + I_{xy} \\ I'_{xy} = I'_{x-1,y} + S_{xy} \end{cases} \quad (2)$$

with $S_{x,0} = 0$ and $I'_{0,y} = 0$.

We will see next how these computations become unnecessary as long as a flexible programmability of the local interconnections of PEs in a focal-plane processing array is provided. By means of this programmability, we will be able to extract the targeted information about a great deal of rectangular blocks in parallel, no matter their scale and location. And that information will not be only constrained to the DC component. Instead, a summary of the different frequency components of each block is achieved, allowing for distinguishing blocks that would look exactly the same if only the DC component is compared.

III. Focal-plane processing scheme

Consider Fig. 2. It represents a focal-plane processing array where each pixel, that is, each photosensor, has a PE associated. Every PE will receive the same instruction to be applied to its corresponding pixel. The point here is that the local

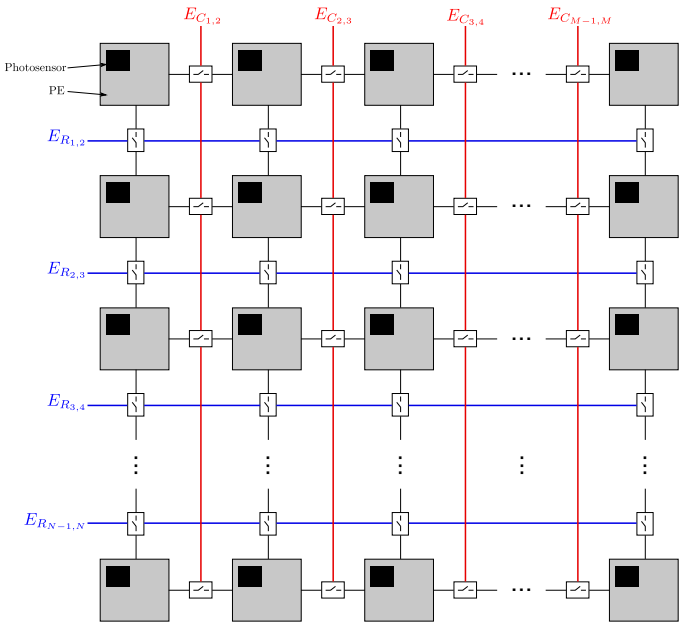

Fig. 2. Focal-plane processing scheme proposed. The PE interconnections can be enabled row-wise and column-wise.

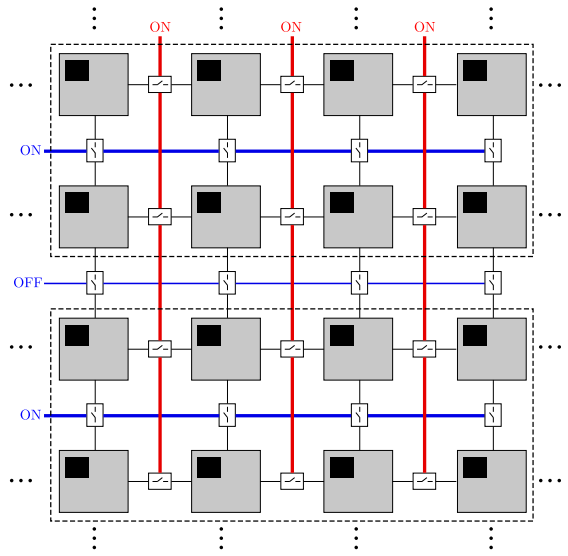

Fig. 3. Example of focal-plane interconnection patterns to generate rectangles like those of Fig. 1(b).

result after applying this instruction can be merged with a programmable neighborhood. These merging regions are set row-wise and column-wise via the interconnection patterns $E_{R_{k,k+1}}$ and $E_{C_{l,l+1}}$ respectively. Thus, when the signal $E_{R_{k,k+1}}$ is ON, all the PEs belonging to the consecutive rows k and $k+1$ combine their results. Similarly, when the signal $E_{C_{l,l+1}}$ is ON, all the PEs belonging to the consecutive columns l and $l+1$ are linked. Otherwise, the corresponding rows and columns remain isolated. Thanks to this programmability of interactions among PEs, any rectangular-shape grouping can be established across the array. An example is depicted

in Fig. 3, where the interconnection patterns render 4×2-px rectangle-shape regions similar to those of Fig. 1(b).

The processing scheme of Fig. 2 has been implemented in [11]. The reconfigurability of the focal plane is realized through two shift registers, each determining respectively the row and column interconnections. These registers enable an easy and ultra fast reconfiguration of the blocks by adequately shifting the bit strings loaded into them. Besides, only four pins — two for the column register and two for the row register — suffice to externally define the focal-plane division. Regarding the elementary PE, one of the processing primitives implemented is the computation of the pixel energy. When the energy of all the pixels within a certain prescribed block is combined, the array delivers the block energy. That is, assuming a rectangle-shape block, no matter its scale and location, labeled as (p, q), we can read out from the chip a value proportional to:

$$E_{pq} = \sum_i \sum_j |W_{ij}|^2 \tag{3}$$

where we have considered, without loss of generality, that we are dealing with one of the white blocks depicted in Fig. 1. It is well known that E_{pq} can be also expressed in the Fourier domain as:

$$E_{pq} = \sum_u \sum_v |\hat{W}_{uv}|^2 \tag{4}$$

where $\hat{W}_{uv}$ represents the different frequency components of the block. In order to demonstrate why this computation of the block energy enables richer and more useful features than those simply based on the sum of pixel values, consider Fig. 4. In the upper rectangle, all the pixels are, alternatively, white and black, the extremes of the signal range. In the lower rectangle, all the pixel values coincide just at the middle of that signal range. If we make a comparison of these two blocks on the grounds of the sum of pixels, both of them look similar as both sums match. However, if the comparison is based on the energy, the upper rectangle features a greater energy because of the high-frequency component that is not present in the lower rectangle. Keep in mind that the DC component of both rectangles is exactly the same, being this the only component for the lower rectangle.

We have therefore defined a computation that permits to gain insight about the content of the blocks over which it is realized. This computation takes only about 225ns to be obtained in the chip reported in [11]. And, more importantly, thanks to its massively parallel operation, that is the time required for the computation at all the focal-plane blocks previously established. Finally, the necessary energy measured for each operation consisting of reconfiguring the focal plane and computing in parallel the energy of all the blocks set is 5μJ. This demonstrates the excellent performance achievable for a SIMD-based array implementing simple, though meaningful, pixel-level operations whose outcomes can be merged with a programmable neighborhood.

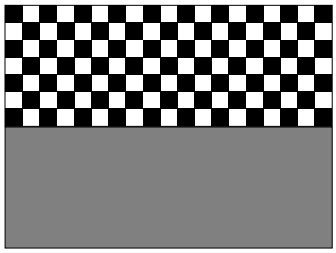

Fig. 4. From the point of view of the sum-based feature extraction, these two rectangles do not make any difference. On the contrary, the energy-based extraction highlights the existence of a high-frequency component at the upper rectangle.

IV. Experimental results

Fig. 5 summarizes experimental results attained with [11]. The first row shows a 176×144-px image captured with the chip and three different focal-plane representations of this image in which the energy is computed for 4×8-px vertical rectangles, 8×4-px horizontal rectangles and 4×4-px square blocks respectively. The procedure to achieve any of these representations, once the original image has been captured, consists of two steps. First, the corresponding focal-plane division scheme is set by loading the adequate interconnection patterns in the shift registers. Second, the energy of the so established blocks is computed in parallel by activating the corresponding control logic at every PE across the array. Once the resulting representation is read out, these two steps are repeated for the next configuration. Note that, according to Fig. 1, each of the proposed division schemes could be useful for Viola-Jones feature extraction within different detection windows. The second row in Fig. 5 includes the same original image as in the upper row but the subsequent energy-based representations were obtained off-chip with MATLAB. While the on-chip operation makes use of Eq. 3 to compute the energy, we have applied Eq. 4 in these off-chip representations, demonstrating thus experimentally their equivalence. Finally, the last row represents the error, normalized in each case to the highest measured error on an individual block, which are, respectively, 39.21%, 26.27% and 27.81%. Despite this large error at certain blocks, the RMSE for each on-chip representation is only 5.82%, 4.71% and 4.57%, respectively. Interestingly, the RMSE decreases as the size of the blocks increases because of the way the focal-plane processing is realized. The energy of all the pixels within a block is added up and averaged to compute the energy of the block. Consequently, the more the number of pixels in a block, the more probable to compensate computation errors during the averaging.

V. Conclusions

We have described how a reconfigurable focal-plane processing array can speed up the execution of the Viola-Jones algorithm by taking advantage of its massively parallel operation. The image plane reconfigurability is achieved by a user-definable selection of row-wise and column-wise PE intercon-

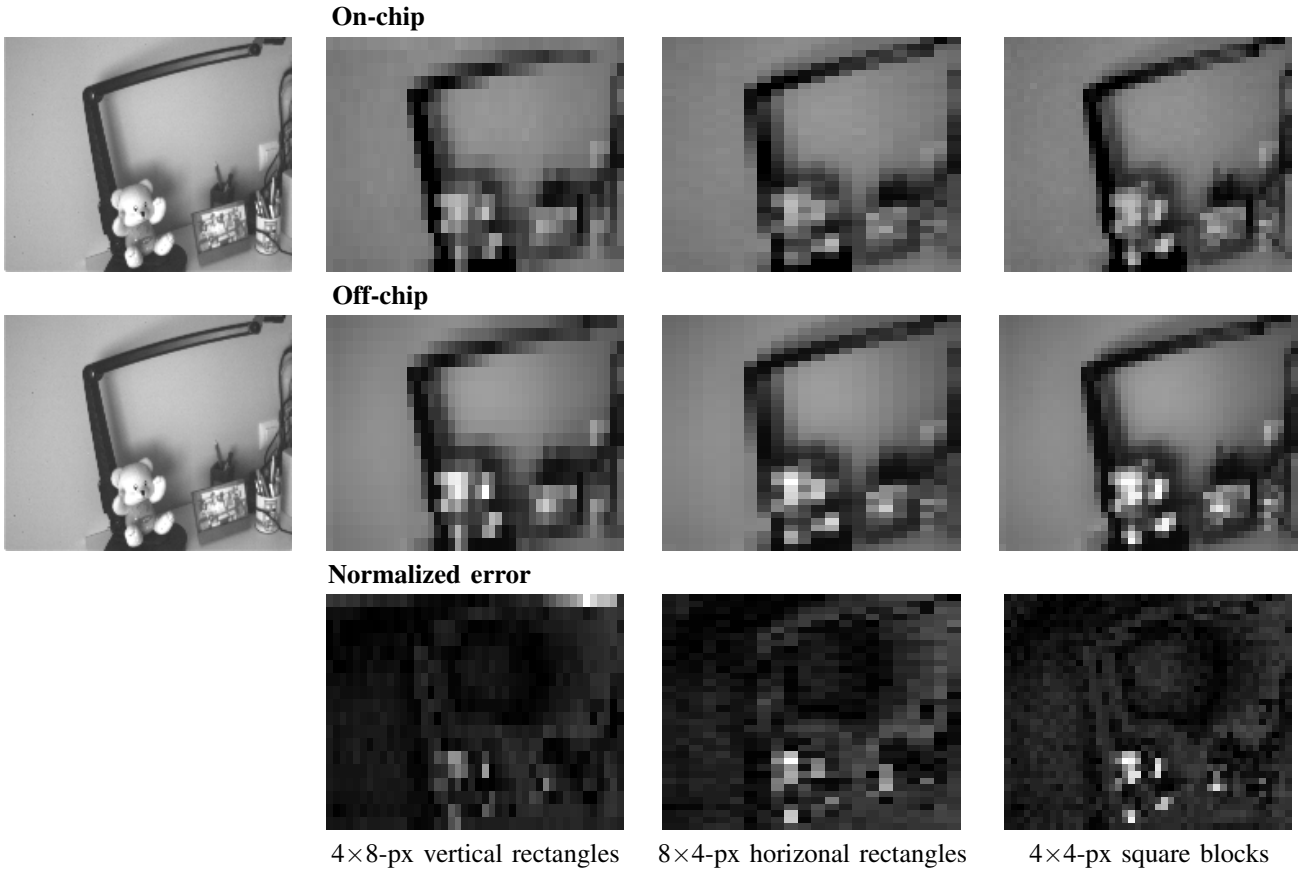

Fig. 5. The first row shows on-chip outcomes: the 176×144-px original image and different block-wise energy-based image representations. The second row includes the same original image and the corresponding off-chip image representations obtained with MATLAB. Finally, the last row represents their normalized difference.

nections. Rectangular-shape blocks can be thus programmed in such a way that the energy of each block instead of its sum is computed in parallel. This computation, which takes only 225ns and 5μJ for all the blocks configured, provides richer information than the sum of pixels, highlighting differences between blocks undistinguishable if the sum is exclusively used. The definition of new features making the most of this enriched information could also accelerate the decision time of the subsequent classifiers.

ACKNOWLEDGMENT

This work is funded by MICINN (Spain) through project TEC2009-11812, co-funded by the European Regional Development Fund, also by Junta de Andalucía under the Project P08-TIC-03674 and by the Office of Naval Research (USA) through grant N000141110312.

REFERENCES

[1] P. Viola and M. Jones, "Robust real-time object detection," in *2nd Int. W. on Statistical and Computational Theories of Vision*, Vancouver, Canada, 2001.

[2] C. Gao and S. Lu, "Novel FPGA-based haar classifier face detection algorithm acceleration," in *IEEE Int. C. on Field Programmable Logic and Applications*, Heidelberg, Germany, 2008, pp. 373–378.

[3] H. Ngo, R. Rakvic, R. Broussard, and R. Ives, "An FPGA-based design of a modular approach for integral images in a real-time face detection system," in *SPIE Mobile Multimedia/Image Processing, Security, and Applications*, Orlando, FL, USA, 2009.

[4] J. Harvey, "GPU acceleration of object classification algorithms using NVIDIA CUDA," Ph.D. dissertation, Kate Gleason College of Engineering, Rochester, NY, USA, 2009.

[5] D. Hefenbrock, J. Oberg, N. Thanh, R. Kastnert, and S. Baden, "Accelerating Viola-Jones face detection to FPGA-level using GPUs," in *18th IEEE Int. S. on Field-Programmable Custom Computing Machines*, Charlotte, NC, USA, 2010, pp. 11–18.

[6] A. Rowe, A. Goode, D. Goel, and I. Nourbakhsh, "CMUcam3: An open programmable embedded vision sensor," Robotics Institute, Carnegie Mellon University, Tech. Rep., 2007.

[7] M. Camilli and R. Kleihorst, "Demo: Mouse sensor networks, the smart camera," in *5th ACM/IEEE Int. C. on Distributed Smart Cameras*, Ghent, Belgium, 2011.

[8] L. Acasandrei and A. Barriga-Barros, "Accelerating Viola-Jones face detection for embedded and SoC environments," in *5th ACM/IEEE Int. C. on Distributed Smart Cameras*, Ghent, Belgium, 2011.

[9] A. Zarándy, Ed., *Focal-plane Sensor-Processor Chips*. Springer, 2011.

[10] S. Unger, "A computer oriented toward spatial problems," *Proceedings of the IRE*, vol. 46, no. 10, pp. 1744–1750, 1958.

[11] J. Fernández-Berni, R. Carmona-Galán, and L. Carranza-González, "FLIP-Q: A QCIF resolution focal-plane array for low-power image processing," *IEEE J. of Solid-State Circuits*, vol. 46, no. 3, pp. 669–680, 2011.

[12] C. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *6th Int. C. on Computer Vision*, 1998, pp. 555–562.